EUROPEAN PATENT APPLICATION

(72) Inventor: **Kovalick, Albert W.**
2650 Webster Court
Santa Clara, CA 95051(US)
Inventor: **Hassun, Roland**
875 Regent Court
San Carlos, CA 94070(US)

(74) Representative: **Schoppe, Fritz, Dipl.-Ing.**
Seitnerstrasse 42
W-8023 Pullach bei München(DE)

(54) Digital synthesizer with phase memory.

(57) In direct digital synthesizers in the prior art, the output signal maintains a phase continuity whenever it switches, or hops, frequency. This phase continuity shows up as a smooth change in phase from one frequency to the next; the phase of the last frequency transitions into the phase of the new frequency without any discernable disruption. Thus, whenever the output signal returns to a switched frequency that it previously has, the output signal at the newly returned switched frequency has a new phase relative to its previous one at that frequency. For some applications, like simulating continuous different frequency sources, this phase continuity is not desirable. To overcome this disadvantage, the preferred embodiment of the present invention provides phase memory to a direct digital synthesizer so that regardless of the frequency that the output signal switches to, the output signal at that frequency is able to maintain a constant phase relative to a reference system clock pulse. In other words, switching and returning the synthesizer output signal to a selected frequency does not change the phase of the output signal at that frequency; the output signal at each frequency has the same constant phase as it has previously at that frequency. As a result, the synthesizer in accordance with the present invention appears to have a source of numerous continuous oscillators to which it can switch for a number of different frequencies, each having a constant phase.
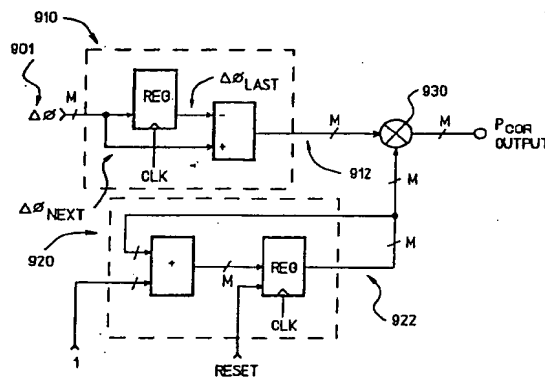
FIG 9A

In the art of synthesizing signals, three distinguishable techniques have been used: direct analog synthesis, indirect analog synthesis, and direct digital synthesis.

In the direct analog synthesis technique, the desired signal is produced by combining and mixing various combinations of signals derived from multiplying and dividing a reference frequency signal. In synthesizing a signal having a wide frequency range, this direct technique becomes extremely complex and costly because of the many components required for the multiplying-dividing and mixing-combining processes. Hence this technique is not widely used for synthesizing signals in either the high frequency or wide frequency ranges.

In indirect analog synthesis, phase lock loops with programmable frequency dividers are commonly used to synthesize a signal with the desired frequencies. This technique is by far the most widely used at present in both commercial products and dedicated applications. The method owes its popularity in large part to the advent of inexpensive programmable frequency dividers in integrated circuit form. The result has been a substantial reduction in complexity, especially in comparison with direct analog synthesis.

Direct digital synthesis is useful for avoiding the above problems associated with analog syntheses. Basically, digital synthesis consists of generating a stream of points with digital logic circuits to represent a desired signal. Then this numerical data stream is converted into the actual desired signal by means of a k-bit digital-to-analog converter (DAC). The DAC output can be further processed to provide a more useful signal. For example, it can pass through a low pass filter for a smoother and cleaner signal and an amplifier for compensating losses incurred through the conversion and filtering processes. The resulting analog output is, ideally, free of sampling components. An example of such a system for synthesizing signals in the prior art is described in U.S. Patent No. 3,928,813. Figure 1 shows a diagram of one such typical digital numerical synthesizer known in the prior art. In this system, the defining relationship for frequency is:

$$F = (\Delta\phi/\Delta T)/(2\pi),$$

where $\Delta\phi$ is the input phase increment for uniquely determining the frequency of the system output signal F and ΔT is the period of the system digital clock, or $1/F_{clock}$, where F_{clock} is the frequency of the system clock. In this system, 2π is defined as 2^M, where M is the number of bits in the field for $\Delta\phi$, which ranges from 0 to 2^M-1.

The basic system in Figure 1 is easily modified to include phase modulation by adding in a phase modulation term, PM(T), at the phase accumulator output 101. Also, frequency modulation may be realized by adding in a carrier phase increment and a frequency modulation phase increment to form the instantaneous phase increment $\Delta\phi$ 102. For a truly universal synthesizer, amplitude modulation may also be added to the system. For this implementation, a fast multiplier may be added at the sine table output 103. Other similar implementations for modulating the system output signal are described in U.S. Patent No. 4,331,941.

The system in Figure 1 may also be used as a fast switching frequency source. Such a usage of this circuit is quite common, because the circuit output 104 can change from F_1 to F_2 just by changing the carrier phase increment 102 from $\Delta\phi_1$ to $\Delta\phi_2$. Because of the complex multiplication and mixing processes associated with each switched frequency that need to be implemented, building a comparably fast switching analog-based synthesizer, however, would be very difficult.

A characteristic of the synthesizer of Figure 1 is its phase continuity between adjacent output frequencies. In other words, the transition from a first frequency F_1 to a second frequency F_2 is accomplished without any abrupt carrier amplitude change. This lack of an abrupt change implies that no phase discontinuity occurs at the instant the system output signal switches from frequency F_1 to frequency F_2. To appreciate this phase continuous nature of the synthesizer output signal, one should consider the synthesizer represented by the phase ramps 201, 202, 203 in Figure 2. The phase ramps 201, 202, 203 form an exemplary output of the phase accumulator 105 of Figure 1 for three different carrier phase increments, $\Delta\phi$ 102: $\Delta\phi_1$, $\Delta\phi_2$ and $\Delta\phi_3$. These carrier phase increments correspond to three output frequencies F_1, F_2 and F_3, respectively. In Figure 2, the phase at each frequency switch point 204 again changes without a discontinuity. In the time increment shown, T_N, a total phase of 66.52 radians has accumulated. This amount is equivalent to approximately 10.5 cycles of a sine wave.

Figure 3 shows the sine wave resulting from the signal represented by the phase accumulation in Figure 2 passing through the sine-lookup ROM 106 and DAC 107 blocks of Figure 1. The transition point for F_1 to F_2 301 and the one for F_2 to F_3 302, in keeping with the non-abrupt changes 204 in accumulated phase, are smooth and phase continuous.

The object of the present invention is not to preserve phase continuity but to allow frequency switching, or frequency hopping, with phase memory, among any number of frequencies F_1, F_2, F_3, . . . , F_N. In other

2

words, with each frequency hop, the output signal assumes a new frequency at the same phase that the signal would have had if the signal had started with the new frequency at a zero common time, T_0, when all the frequencies start at zero phase. Thus, the output of the signal in accordance with the present invention is equivalent to a signal resulting from switching among a series of separate oscillators having frequencies
5   F_1, F_2, . . ., F_N respectively, all of which start at time T_0. Figure 4 shows an example of such an equivalent analog system with phase memory. Three oscillators 401, 402, 403, which run continuously, are locked together to start simultaneously at T_0. A switch 404 is used to select which oscillator is to be the output signal 405. Since all three oscillators 401, 402, 403 never stop, whenever the switch 404 selects a different source, the phase will seemingly jump to account for the current phase of the selected source. This system
10   always has as its output 405 a source with its correct phase at the instant of switching. This ability of a system to maintain and remember a constant phase for each switched frequency as though each has started at a common starting time and be able to switch into any new frequency at the appropriate point of the constant phase is termed "phase memory." Figure 5A shows the output signal of the system in Figure 4 as it switches among F_1, F_2 and F_3. The "glitch" 502 that occurs at the instant of switching 504 is
15   characteristic of a frequency hopped source with phase memory.

Figure 5B shows the phase term of the signal in Figure 5A; it is analogous to the phase term in Figure 2. The phase is composed of three different phase slopes 520, 522, 524. The first slope 520 starts at $T = 0$ and corresponds to F_1. The other two slopes 522, 524 correspond to F_2 and F_3. In fact, all three started their rise at $T = 0$, but only one oscillator is sampled for an actual output at any given time. Figure 5C
20   illustrates the phase of all three oscillators simultaneously. Figure 5B also shows the phase offsets $P_{Cor1\rightarrow2}$ 526 and $P_{Cor2\rightarrow3}$ 528 that exist when a frequency hop occurs from F_1 to F_2 and from F_2 to F_3, respectively; they are the cause of the abrupt change 502, or "glitch," during its frequency transition 504.

The prior art system in Figure 1 may be modeled as a single oscillator that may be changed to a new frequency only by changing the value of the input phase increment $\Delta\phi$ 102. The starting phase of the new
25   frequency is just the ending phase of the one that preceded it. The present invention extends the system in Figure 1 to appear that many oscillators are all running simultaneously and that only one is being sourced as the output at any given time. Since both phase memory and phase continuity are aspects of frequency hopping systems, one of the two is usually required and is determined by the exact application of the frequency hop.
30

Summary of the Invention

To overcome the lack of a phase memory in prior art direct digital synthesizers, the present invention detects a frequency hop in response to a change in input phase increment and calculates a phase
35   correction signal. In accordance with the present invention, the preferred embodiment uses a difference operator to determine the change, or difference, in the phase increment input. Then this difference value is combined, for example, through a multiplier with the output from a counter which accumulates a timing reference signal to produce the phase correction signal. The phase correction signal in turn is combined with the synthesized digital signal representing the desired output signal at a selected frequency, F_s. The
40   resultant output signal at that selected frequency then maintains a constant phase relative to a reference system clock pulse for the duration of the synthesizer operation. Consequently, regardless of the number of intervening frequency hops the output signal may undergo and regardless of the frequencies it switches to during this intervening period, whenever the output signal switches to frequency F_s, the output signal at frequency F_s has the same constant phase it has at that frequency as before. If the output signal switches
45   to a newly selected frequency, the preferred embodiment of the present invention will provide the output signal with the same constant phase if the output signal were to switch to that frequency at a later time.

Brief Description of the Drawings

50   Figure 1 illustrates a digital signal synthesizer of the prior art.
Figure 2 shows the phase accumulator output having three different carrier phase increments.
Figure 3 shows the sine signal of the phase accumulator system in Figure 1.
Figure 4 illustrates a system multiplexing three oscillators to arrive at a synthesized output signal.
Figure 5A shows the abrupt changes at the points of transition in the multiplexed system of Figure 4.
55   Figure 5B shows the equivalent phases of the output signal in a system like the one in Figure 5A.
Figure 5C illustrates the phases of three oscillators all starting at $T = 0$.
Figures 6A-6D are timing diagrams of the phase corrections relative to a frequency hop for phase memory.

3

Figures 7-8 illustrate examples of a system with phase memory in accordance with the invention.

Figure 9A illustrates the preferred embodiment of a phase correction scheme to achieve phase memory in a digital synthesizer system in accordance with the invention.

Figure 9B contains a diagram of the carrier phases in a system with phase memory switching among
5   frequencies F_1, F_2 and F_3.

Figure 10 shows a preferred embodiment of the invention to allow the use of slower multipliers.

Detailed Description of the Invention

10   The present invention enhances and extends the capability of the prior art synthesizer by including a phase memory in it. Specifically, the preferred embodiment of the invention has the following characteristics:

1. At time T = 0, all "oscillators," that is, the non-existent virtual "oscillators" that provide the various frequencies for the system to switch to, are simultaneously "started." By definition, all of them start with
15   an initial phase of zero radian.

2. With a phase accumulator of M-bits, there are $2^M/2$ frequencies that may be generated by a prior art synthesizer like the one featured in Figure 1. The preferred embodiment of the present invention will maintain phase memory for each $2^M/2$ frequencies. Through simplification in design, however, it is possible to provide phase memory for fewer frequencies, if desired, with a concomitant saving in
20   manufacturing costs.

3. The invention includes a circuit that calculates a phase correction value, P_{Cor}, which is permanently added to the carrier phase to enable a signal of a first frequency F_1 to hop or switch to a second frequency F_2 with the proper phase offset so that the signal looks as it it had never left the second frequency from its inception at T = 0, thus resulting in phase memory.

25   4. Once the synthesizer in accordance with the preferred embodiment has started (at T = 0), it never needs to be reset or resynchronized at any time to align the phases of each of the $2^M/2$ frequencies for phase memory. In other words, the system in accordance with the preferred embodiment of the invention will feature phase memory with no degradation of performance for an indefinite number of frequency hops.

30   These characteristics will become more evident as the mathematical basis of the present invention is discussed.

If N is the number of clock pulses having a period ΔT that have elapsed, the phase at time $N \cdot \Delta T$ of a carrier of frequency F_i is:

35

$$\text{Phase of } F_i = 2\pi \cdot F_i \cdot N \cdot \Delta T \text{ Modulo}(2\pi)$$

$$= \omega_i \cdot N \cdot \Delta T \text{ Modulo}(2\pi).$$

40
Also, the phase of F_j at time $N \cdot \Delta T$ is:

Phase of $F_j = \omega_j \cdot N \cdot \Delta T$ Modulo(2π).

45   Any of the possible $2^M/2$ frequencies to which an M-bit digital synthesizer may be switching may be represented this way.

If F_j is the frequency of the current output signal of the synthesizer and F_i is the new frequency the output signal is being switched to, then the phase correction P_{Cor} needed for phase memory at time $N \cdot \Delta T$ is:

50

$$P_{Cor} = [\omega_i \cdot N \cdot \Delta T - \omega_j \cdot N \cdot \Delta T] \text{ Modulo}(2\pi)$$

$$= (\omega_i - \omega_j) N \cdot \Delta T \text{ Modulo}(2\pi).$$

55

This last equation is interpreted as the difference between two phase ramps. It is this difference term in P_{Cor} that must be permanently added to the carrier phase of the synthesizer for phase memory in accordance

4

with the invention to take place.

The expression and hence the implementation of phase correction P_{Cor} may be simplified. From Figure 1, one sees that:

5   $F_i = (\Delta\phi_i/\Delta T)/(2\pi)$.

Then:

$2\pi F_i = \omega_i = \Delta\phi_i/\Delta T$, and

10   $P_{Cor} = (\Delta\phi_i \cdot N - \Delta\phi_j \cdot N)\text{Modulo}(2\pi) = (\Delta\phi_i - \Delta\phi_j) \cdot N \ \text{Modulo}(2\pi)$.

In other words, the phase correction is the difference in accumulated phase increments with modulo 2π at the Nth clock pulse. Also, as shown in the system of Figure 1, $2\pi \equiv 2^M$. Therefore, the above equation for P_{Cor} can also be expressed as follows:

15   $P_{Cor} = (\Delta\phi_i - \Delta\phi_j) \cdot N \ \text{Modulo}(2^M)$.

By extension, the phase correction P_{Cor} needed when switching to a new frequency F_k from an existing frequency F_j is:

20   $P_{Cor} = (\Delta\phi_k - \Delta\phi_j) \cdot N \ \text{Modulo}(2^M)$.

Thus, the phase correction P_{Cor} is easily calculated so long as the current phase increment $\Delta\phi_j$ and the next phase increment $\Delta\phi_k$ are known. The phase correction is calculated at the instant the current phase

25   increment changes to the next one. Also, the phase correction must, by some means, be stored currently and used subsequently to offset the carrier phase permanently.

Figures 6A through 6D show the timing sequence of a signal having three frequency hops. Figure 6A shows the clock pulses of a system clock having a frequency F_{clock}. Figure 6B shows three differing input phase increments $\Delta\phi_1$, $\Delta\phi_2$ and $\Delta\phi_3$. Figure 6C shows the corresponding phase corrections at the instant

30   of the frequency hops. And Figure 6D shows the phase corrections being accumulated to provide phase memory to the output signal of the system. The values of $\Delta\phi$ correspond to those required for the synthesizer in Figure 5A. Thus,

35   $\Delta\phi_1 = 3.4 \ (2\pi)/200$ (1.7% of F_{clock})

$\Delta\phi_2 = 1.2 \ (2\pi)/200$ (0.6% of F_{clock})

40   $\Delta\phi_3 = 5.75(2\pi)/200$ (2.9% of F_{clock}).

The phase correction is easily calculated at the instance of a frequency hop. If each of the three phase increments is of a duration of, for example, 200 clock pulses, then N = 200 at the transition from $\Delta\phi_1$ to $\Delta\phi_2$ and N = 400 at the transition from $\Delta\phi_2$ to $\Delta\phi_3$. Thus, the phase correction for the first transition is:

45

$P_{Cor(1\to2)} = (1/200)(2\pi)(1.2-3.4)200 \ \text{Modulo}(2\pi)$

50   $= (0.2)(2\pi)$

$= -1.25 \text{ radians or } -72°$.

This amount is the phase jump from F_1 to F_2 that the signal takes in Figure 5A. Similarly, the phase
55   correction for the transition from F_2 to F_3 is:

$P_{Cor(2\to3)} = 0.628$ radians.

Since every phase correction value must be permanently added to the carrier phase, these values must be stored in a P_{Cor} phase accumulator. This P_{Cor} phase accumulator contains the composite sum of every previously computed phase correction value. Figure 6D illustrates the accumulated sum of all previous phase corrections for the frequency hops of the output signal in Figure 5A. This sum must be used to offset the carrier phase. In effect, every new phase correction value must be added to the past sum of every other phase correction value.

Phase corrections in accordance with one embodiment of the invention lasts only for a clock pulse and occurs immediately after a frequency hop. This fact is illustrated by the timing diagram of Figure 6C. The placement of a phase correction calculator 701 is shown in the example system of Figure 7. In this system, a phase correction phase accumulator 702 for retaining the phase corrections 703 and maintaining an accumulated sum of them 706 is also shown. A phase offset adder 708 adds the accumulated sum of phase corrections 706 to the accumulated carrier phase 710 produced by a carrier phase accumulator 712. An alternate preferred embodiment of the invention is illustrated in Figure 8. This embodiment, though achieving identical results as the system in Figure 7, does not require a phase accumulator for phase corrections 702. Instead, with each change of the carrier phase increment 810, the resultant phase correction P_{Cor} 812 is added to the phase increment $\Delta\phi$ 810 for one clock period by a phase correction adder 802. This nonzero value of P_{Cor} 812 lasts only one clock cycle and is added into the carrier phase accumulator 804 by the input adder 806 of the carrier phase accumulator 804. In effect, the phase correction offset is permanently added to the carrier phase. To account for a possible delay of the P_{Cor} signal 812 from the input phase increment 810 $\Delta\phi$ as the P_{Cor} signal 812 propagates from the P_{Cor} calculator 801, a delay block 815 serves to delay the phase increment $\Delta\phi$ 810 before it enters the P_{Cor} adder 802. This delay ensures that when a new input phase increment $\Delta\phi$ 810 is present, the P_{Cor} output signal 812 produced by the P_{Cor} calculator 801 in response to the new phase increment $\Delta\phi$ 810 will be coincident with the new input phase increment $\Delta\phi$ 810 at the input ports of the P_{Cor} adder 802.

As shown above, the expression for a phase correction offset is as follows:

$$P_{Cor} = (\Delta\phi_{next} - \Delta\phi_{last}) \cdot N \ \text{Modulo}(2^M).$$

This expression is implemented by the circuit illustrated in Figure 9A. Figure 9A has three main blocks: a difference operator 910, a counter 920 for accumulating the number N of clock cycles that have elapsed as a timing reference, and a multiplier 930. The difference operator 910 computes to M-bits the difference in phase increments 912 during a frequency hop, $(\Delta\phi_{next} - \Delta\phi_{last})$. This value is zero except during the one clock period following a change in the phase increment 901. This value can range from 0 to 2^M-1. The N counter 920 is the timing reference. It is reset to zero at T = 0 by external means (not shown). Its output 922 ramps from 0 to 2^M-1 and then repeats, because of its modulo 2^M operation. The preferred embodiment of the invention places no restriction on the size of N. However, the N counter 920 is designed as a modulo device with a modulo of (2^M) to take advantage of the implementation of a modulo phase correction calculator, since a phase correction P_{Cor} can also be expressed as follows:

$$P_{Cor} = A \cdot B \ \text{Modulo}(2^M)$$

$$= [A \ \text{Modulo}(2^M)] \cdot [B \ \text{Modulo}(2^M)] \ \text{Modulo}(2^M).$$

This relationship follows from the decomposition theorem in number theory. Carrying this relationship farther, the phase correction P_{Cor} can also be expressed as:

$$P_{Cor} = [(\Delta\phi_{next} - \Delta\phi_{last})\text{Modulo}(2^M)] \cdot [N \ \text{Modulo}(2^M)]\text{Modulo}(2^M).$$

This form of the phase correction allows the multiplier 930 in Figure 9A to have an output of modulo(2^M) also. Thus, all three blocks 910, 920, 930 in Figure 9A function in a modulo(2^M) mode. In practical terms, the output of any of the blocks 910, 920, 930 in Figure 9A uses only the lowest M-bits. Modulo math arises because when dealing with carrier phase, only values in the range of 0 to 2π, that is, 0 to 2^M, are important.

As an example, suppose the preferred embodiment of the invention as depicted in Figures 8 and 9A has the following values:
M bits = 30 bits

L significant bits = 12 bits
System clock frequency F_{clock} = 134.217 MHz
System output signal frequency F_{out} range = 0 to 67.1 MHz
Output signal frequency F_{out} step resolution

5   = $(1/2^{30})(134.217)$ MHz
= 0.125Hz
Phase correction P_{Cor} resolution
= $(2\pi)/(2^{30})$ radians
= 5.85 E-9 radians

10 The carrier phase accumulator output **814** is then truncated to L = 12 bits. The actual phase accuracy obtained when a frequency hop occurs therefore is $(2\pi)/(2^{12})$ = $(2\pi)/4096$, or 0.00153 radians. The phase correction components of Figure 9A, however, is preferably comprised of 32-bit logic/arithmetic elements. As noted previously, the frequency resolution for this example is 0.125 Hz. If the desired frequency resolution could be larger, then the phase memory circuitry may have a concomitant reduction in bit width.

15   As described thus far, the preferred embodiment of the invention can accept every pulse of a clock and calculate a phase correction P_{Cor} for the sum of the pulses accumulated up to the current pulse. If the phase correction calculator logic block is pipelined with clocked registers as shown in Figure 9A, the frequency hop rate could be as high as 100 MHz with a proper circuit design. For most applications, however, a frequency hop rate as high as 100 MHz is definitely not needed. In fact, in most current practical

20 systems, the hop dwell time is generally greater than 250 nanoseconds.

    In systems that only need to hop (or switch frequency) every N_{clock} cycles, it is possible to simplify the phase correction logic. With this simplification, less expensive, and hence more accessible, multipliers can be used. Figure 10 illustrates this simplification. If the phase increment $\Delta\phi$ is constrained to changes no faster than N_{clk} cycles of the clock, then the multiplier **1010** has N_{clock} cycles to do the multiplication. The

25 multiplier **1010** is isolated through holding registers **1012, 1014, 1016** from any inputs changing. A hold command **1020** is generated by a phase increment change detector **1022**, which detects a nonzero output from a difference operator **1024**. The difference operator **1024** detects a change in the input phase increment; a nonzero output from it **1024** starts the multiply cycle. Timing for this simplified circuit is straightforward. Whenever the phase increment $\Delta\phi$ **1026** changes, the value **1030** in an N counter **1028** for

30 accumulating a time count by counting the system clock pulses is held in a first register (Register A) **1014** and the difference in phase increments from the last one to the next one, that is, $\Delta\phi_{next}-\Delta\phi_{last}$, is held in another register (Register C) **1012**. Since a correction signal is not needed until the input phase increment changes again, for example, after N_{clock} cycles, the multiplier **1010** has N_{clock} cycles to produce a stable output. The hold-delay signal **1020** causes a third register (Register B) **1016** to hold the output **1032** of the

35 multiplier **1010** until the multiplication is complete for the phase correction signal, P_{Cor} **1032**. It should be noted that since the phase correction **1032** is delayed, the incoming phase increment $\Delta\phi$ **1026** must also be delayed so that both the new phase increment **1034** and phase correction **1032** are always coincident at the phase correction block output. A delay circuit **1046** is added to the circuitry for this purpose. In sum, in the simplified embodiment of the invention, all logic circuits other than the multiplier operate at the system

40 clock rate F_{clock}.

Claims

1.   A direct digital synthesizer having a system clock with a reference clock pulse and an output signal
45   shiftable to a plurality of frequencies comprising:
    a signal generating means (**[phase accumulator] 712**) responding to a phase increment input for generating said output signal, said phase increment determining one of said plurality of frequencies for said output signal; and
    a phase memory means (**[phase correction calculator] 701**) coupled to said signal generating
50   means (**[phase accumulator] 712**) for generating a correction signal (**[P_{Cor}] 706**) to allow each of said frequencies to maintain a constant phase relative to said reference clock pulse for said generated output signal to shift to.

2.   The direct digital synthesizer of Claim 1, wherein said correction signal of said phase memory means
55   comprises a phase correction signal.

3.   The direct digital synthesizer of Claim 2, wherein said phase correction signal is generated by a phase correction calculator comprising:

7

a difference operator (910, 1024) for determining an increment difference (912, 1044) in said phase increment input (901, 1026;)

a counter (920, 1028) for accumulating a time reference count (922, 1030;) and

a combining means ([multiplier] 930, 1010) coupled to said difference operator (910, 1024) and said counter (920, 1028) for combining said increment difference (912, 1044) and said accumulated time reference count (922, 1030) to produce said phase correction signal .

4. The direct digital synthesizer of Claim 3, wherein said combining means comprises a multiplier.

5. The direct digital synthesizer of Claim 3, further comprising:

a first register (1012) interposed between said combining means (1010) and said difference operator (1024) for holding said increment difference (1044) for a holding period;

a second register (1014) interposed between said combining means (1010) and said counter (1028) for holding said accumulated time reference count (1030) for said holding period; and

one or more delay means (1024) interspersed within said synthesizer for delaying said phase increment input (1026) and said phase correction signal (1032) for a substantially simultaneous output of both.

6. The direct digital synthesizer of Claim 5, wherein said holding period comprises a period starting from a clock pulse with a first increment difference and ending with a clock pulse with a second increment difference.

7. The direct digital synthesizer of Claim 1, further comprising:

an intermediate output (710) from said signal generating means (712);

an adder means (708) for coupling said signal generating means (712) and said phase memory means (701) and for combining said intermediate output (710) with said correction signal (706) to produce an offset phase signal (714); and

a sine means (716) coupled to said adder means (708) for responding to said offset phase signal (714) to produce said generated output signal.

8. The direct digital synthesizer of Claim 1, wherein said phase memory means comprises:

a phase correction calculator (801) for generating said correction signal (812;) and

an adder (802) coupled to said phase correction calculator (801) for producing said phase correction signal for one system clock pulse; and

wherein said phase memory means is coupled to said signal generating means (804) by said adder (802).

9. A method for digitally synthesizing an output signal shiftable to a plurality of frequencies comprising the steps of:

generating an output signal of a *kth* frequency in response to an *ith* phase increment input;

shifting said output signal to a next *kth* frequency in response to a next *ith* phase increment input; and

phase correcting said output signal in response to said next *ith* phase increment input such that said output signal has a constant phase relative to a system reference clock pulse irrespective of *k* or *i*, where *k* is any one of said plurality of frequencies and *i* is any one of a plurality of frequency hops.

10. A method for digitally synthesizing an output signal shiftable to a plurality of frequencies comprising the steps of:

generating an output signal of a first frequency having a constant first phase relative to a system reference clock pulse in response to a first phase increment input;

shifting said output signal to a second frequency in response to a second phase increment input; and

phase correcting said output signal in response to said second phase increment input such that said output signal shifts to said second frequency and said second frequency has a constant second phase relative to said system reference clock pulse.

11. The method for digitally synthesizing an output signal as in Claim 10 further comprising the steps of:

returning said output signal to said first frequency in response to a return to said first phase

8

increment input; and

phase correcting said output signal in response to said first increment input such that said output signal returns to said first frequency and said first frequency has said constant first phase relative to said system reference clock pulse.

5

12. The method for digitally synthesizing an output signal as in Claim 10, wherein the step of phase correcting comprises the steps of:

calculating a phase increment difference between said first and second phase increment inputs;

accumulating a time reference count from said first phase increment input to second phase
10 increment input; and

producing in response to said phase increment difference and said time reference count a phase correction signal for correcting said output signal at said second frequency.

13. The method for digitally synthesizing an output signal as in Claim 11, wherein said step of phase
15 correcting comprises the steps of:

calculating a phase increment difference between said second and first phase increment inputs;

accumulating a time reference count from said second phase increment input to said returned first phase increment input; and

producing a phase correction signal for correcting said output signal at said returned first
20 frequency.

14. The method for digitally synthesizing an output signal as in Claim 9, wherein said step of phase correcting comprises the steps of:

calculating the difference between said *ith* and said next *ith* phase increment inputs;
25 accumulating a time reference count between said *ith* and said next *ith* phase increment inputs; and

producing a phase correction signal for correcting said output signal at said next *kth* frequency.
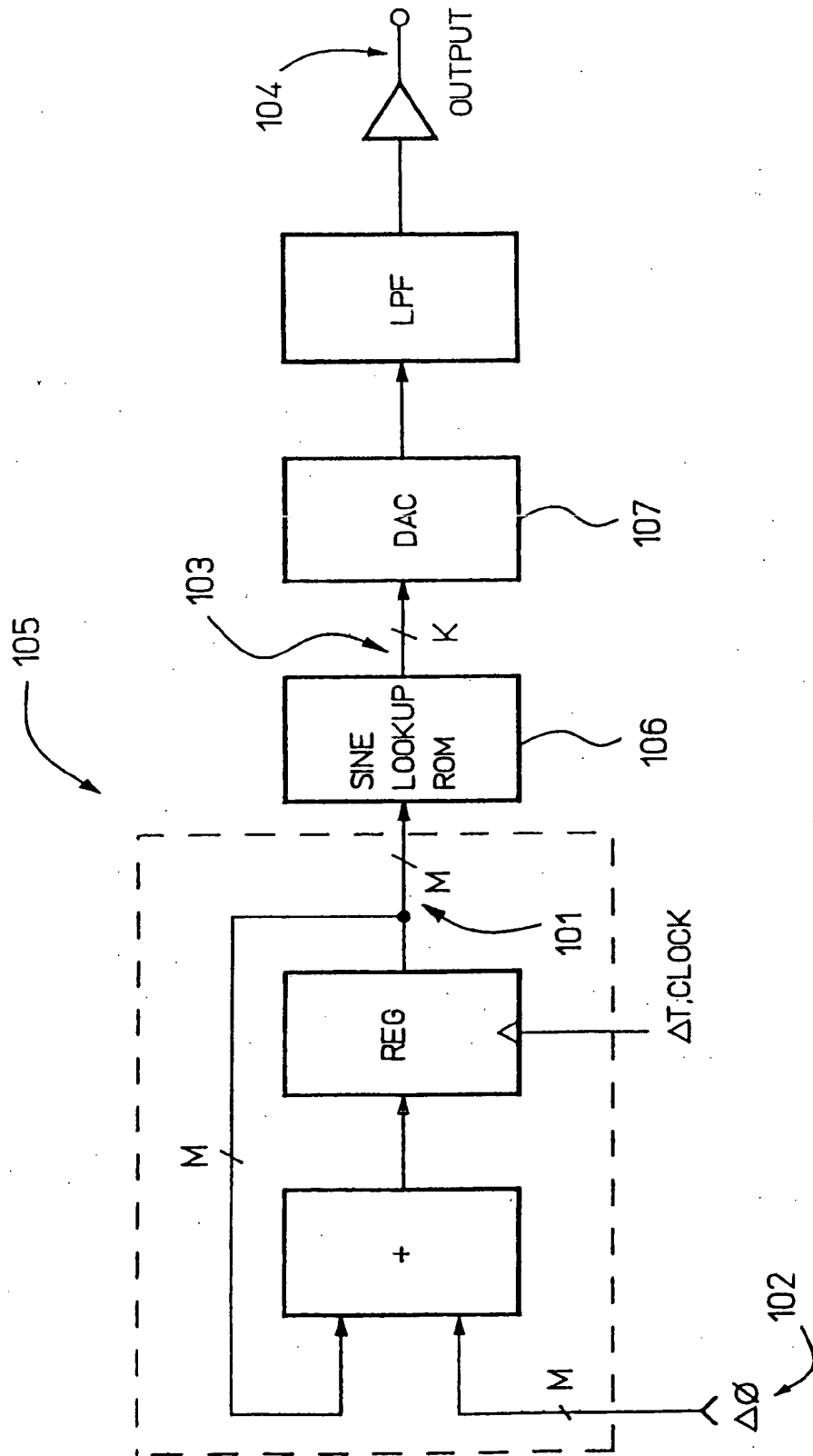
30

35
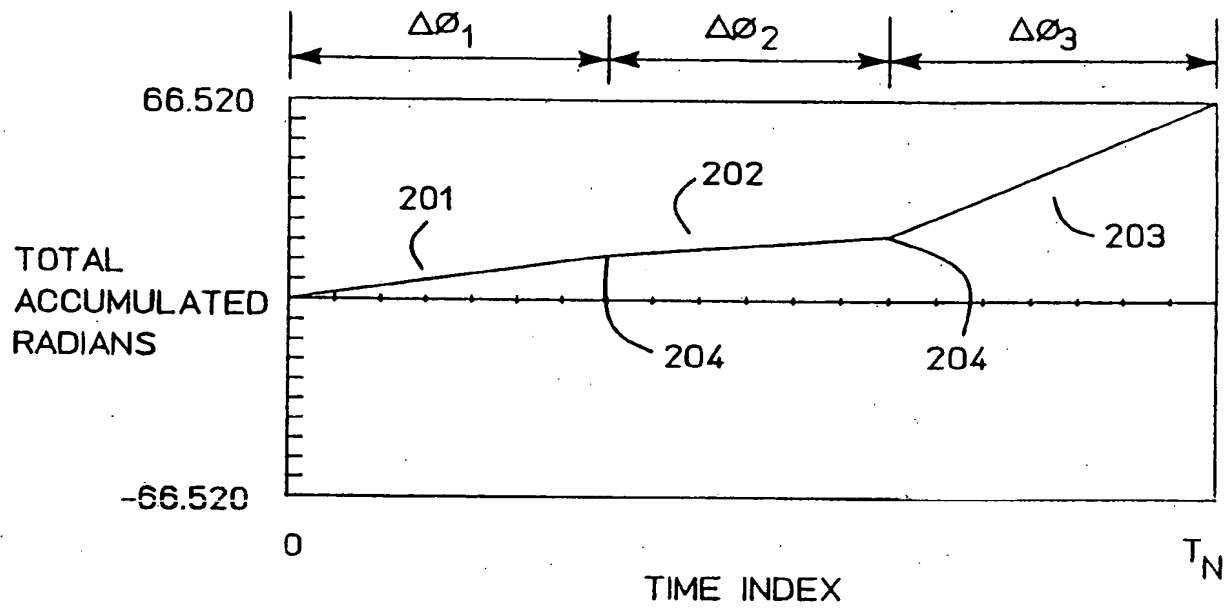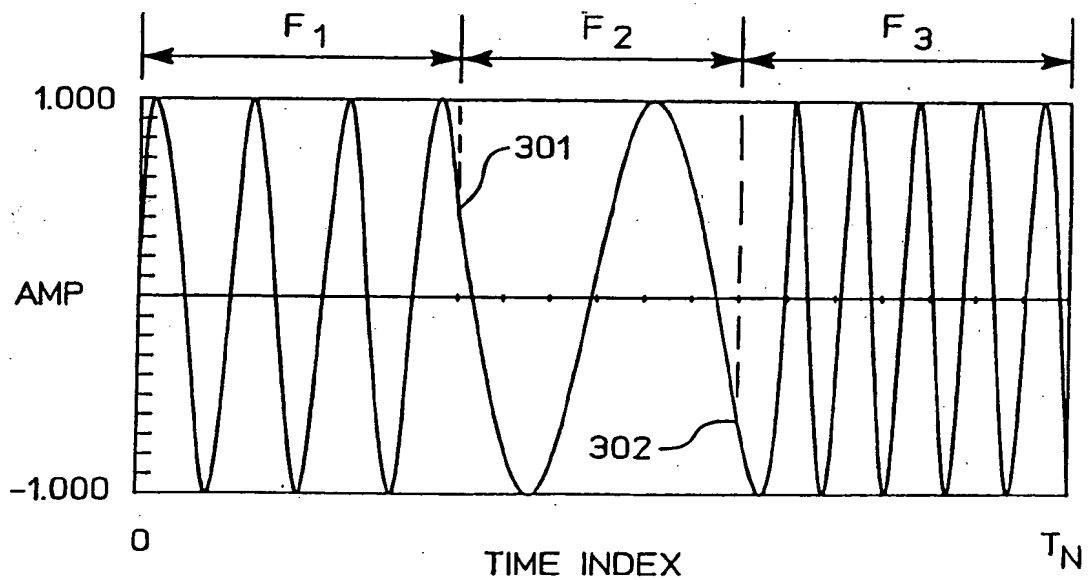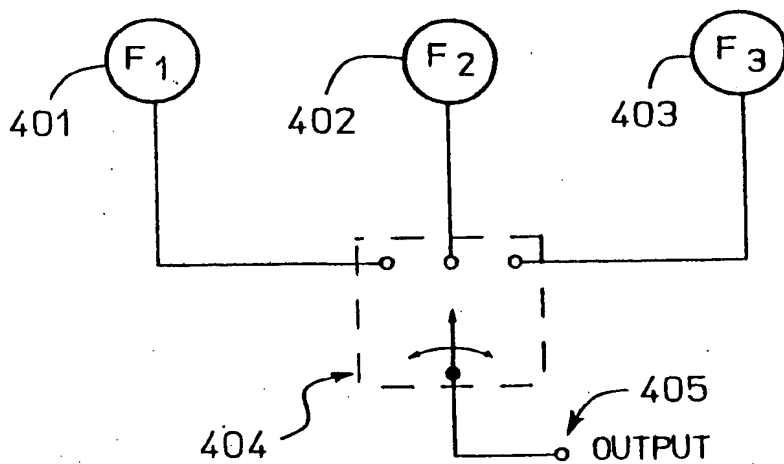
40

45

50

55

FIG 1 (PRIOR ART)

FIG 2

FIG 3

FIG 4

FIG 5A

FIG 5B

FIG 5C

FIG 6A

CLOCK IN

T=0 1 2 3 ≋ 200 201 ≋ 400 401

FIG 6B

Δϕ SOURCE

$\Delta\phi_1$ $\Delta\phi_2$ $\Delta\phi_3$

FIG 6C

P COR (RADS)

ZERO -.125 0.628

FIG 6D

ACCUMULATED P COR VALUES

(LAST) LAST-.125 LAST-.125+.628

FIG 7

FIG 9A

FIG 8

FIG 9B

FIG 10